AMENDMENTS TO THE SPECIFICATION

Please replace paragraph [0002] with the following:

[0002] The usage of the Internet has increased dramatically over the last several years. Many popular websites receive millions of "hits" each day. Consequently, the network servers providing content for these websites have ~~experienxced~~ experienced a dramatic increase in their workload. In order to process such substantial amounts of network traffic without subjecting clients (surfers) to annoying delays in retrieving web pages, it is advantageous to distribute the applications (or services) handling this traffic among multiple web server hardware nodes, so that the multiple server nodes can operate in parallel to process the network traffic.

Please replace paragraph [0030] with the following:

Fig. 2 provides a block diagram illustrating operation of the inventive resource type wizard. Upon the beginning of execution, at block 21, the inventive resource type wizard requests the user to input certain information in block 22, which is used to generate a set of source code files based on the input provided by the user (see block 23). In one embodiment of the invention, the user has the option of having the source code for the resource type generated in C, ksh or perl. C is a programming language well known to persons of skill in the art. Ksh and perl are script programming languages also well known to persons skilled in the art. The aforementioned ksh format is more useful in case the user does not have access to a C compiler or if the user is more comfortable with ksh source code. However, for the ease of extensibility and the richness of underlying functionality, it is preferable to select C as the type of the generated source code. In addition, resource types with ksh source code do not support non-network aware applications or applications with multiple process trees. The inventive wizard

tool also generates and customizes the resource type registration (RTR) file. At block 24, the user has the ability ot modify the generated code. Note that the user can return to block 24 at any time after the code for the resource type has been generated. The generated resource code is installed on nodes of the clustered computer system in ~~block 24~~block 25. The user is allowed to start, stop, and monitor the application at block 26. The operation of the inventive wizard terminates in ~~block 25~~block 27.

Please replace paragraph [0034] with the following:

Using the inventive wizard, each type of application can be wrapped either in the aforementioned failover or scalable resource type. The inventive wizard may also support applications having more than one independent process tree that need to be monitored, and restarted, by ~~PMF~~ a process monitoring facility (PMF) individually. The manner of creation of such resource types, is described in detail below.

Please replace paragraph [0036] with the following:

[0036] Creation of the resource types using the inventive wizard is a process consisting of one or more of the following two steps, wherein the exact order of the steps or their substeps is not critical to the invention:

- In the first step the user is asked to input some basic information about the resource type to be generated. The input information may include, among other parameters, one or more of the following:
  - the resource type name
  - the vendor id: this would typically be the stock symbol of the vendor or some other identifier that uniquely identifies the vendor. This field is initialized to SUNW (Sun®'s stock symbol on NASDAQ®). Sun® is a registered trademark of SUN

Microsystems, Inc. of Santa Clara, California. NASDAQ® is a registered trademark of NASDAQ Stock Market, Inc.

- installation directory: the user specifies a directory under which the wizard will create a subdirectory formed, for example, by the concatenation of the vendor_id and the resource_type_name. For example, if vendor_id is SUNW and resource_type_name is "ftp", then the wizard will create a subdirectory SUNWftp under the installation directory specified by the user. SUNWftp will then contain everything that's generated for the target resource type. The wizard also generates a .rtconfig file to keep track of, and to reuse, the user supplied input. The installation directory is initialized to the directory from where the wizard is started.
- failover of scalable: the user chooses whether the target resource type is failover or scalable.
- network aware: the user specifies whether the base application is network aware, i.e. if it uses the network to communicate with its clients.
- C or ksh : the user selects the language in which the source code is to be generated.

- In addition to, or alternatively to the first step, in the second step, the user may be asked to input various configuration parameters for the target resource type. This may include one or more of the following:
  - start command: This the full command line that can be passed to any unix shell to start the application. The complete command line may include all parameters needed to start the application, such as hostnames, port numbers path to configuration files etc. The completed command line may be placed in quotes.

- stop command: this optional input specifies the command to be run to stop the base application. If this input is omitted, the generated code uses signals to stop the base application.
- probe command: this optional input specifies a command that can periodically be invoked to do a health check on the application and return appropriate exit status (zero for success and non zero for failure). This command would typically be a simple client of the base application. If this input is omitted, the generated code simply connects and disconnects to the port specified and, if that succeeds, declares the application healthy. The probe command can only be used with network aware applications and is disabled for non-network aware applications.
- timeouts: this field may include the timeouts for the start, stop and probe commands. In an embodiment of the invention these timeouts are initialized to 300, 300 and 30 seconds respectively.

A4